UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1272

Programme 1
Programmation, Calcul Symbolique
et Intelligence Artificielle

RAPPORT SUR LE DECODAGE DES CODES GEOMETRIQUES

Dominique LE BRIGAND

Août 1990

RR-1272

RAPPORT SUR LE DECODAGE

DES CODES GEOMETRIQUES

ON THE DECODING OF THE GEOMETRIC CODES

Dominique LE BRIGAND*

(MAI 1990)

***Université Pierre et Marie Curie (Paris 6)**

4 place Jussieu 75250 PARIS Cedex 05

chercheur extérieur - projet CODES

RESUME . En 1989 le décodage des codes géométriques, ou "codes de GOPPA", a beaucoup progressé grâce aux idées de JUSTESSEN et al.. Tout d'abord SKOROBOGATOV et VLADUTS ont donné un algorithme de décodage de complexité $O(n^3)$ qui corrigeait jusqu'à $\lfloor \frac{(d^*-1-g)}{2} \rfloor$ erreurs, où n est la longueur du code, d^* sa distance prévue et g le genre de la courbe. Ensuite PELLIKAAN a amélioré cet algorithme pour corriger jusqu'à $\lfloor \frac{(d^*-1)}{2} \rfloor$ erreurs avec une complexité $O(n^4)$; il a montré que cet algorithme est valable pour des codes géométriques construits à partir de courbes maximales, c'est à dire des courbes dont le nombre de points rationnels sur le corps atteind la borne supérieure de WEIL. Enfin VLADUTS a donné une classe plus large de courbes pour lesquelles l'algorithme est appliquable. Dans ce rapport nous exposons les améliorations successives du décodage en commençant par les idées de JUSTESSEN et al. qui sont en fait une généralisation du décodage des codes de REED-SOLOMON.

ABSTRACT: During the year 1989, the decoding for geometric codes, "GOPPA Codes", has been deeply improved thanks to JUSTESSEN et al. ideas. First SKOROBOGATOV and VLADUTS gave a decoding algorithm with complexity $O(n^3)$ which corrects up to $\lfloor \frac{(d^*-1-g)}{2} \rfloor$ errors, where n is the word length of the code, d^* its designed distance and g the genus of the curve. After that, PELLIKAAN improved the algorithm to correct up to $\lfloor \frac{(d^*-1)}{2} \rfloor$ errors with complexity $O(n^4)$; he showed that the algorithm is available for codes on maximal curves, that is curves for which the number of rational points on the field attains the WEIL upper-bound. Last VLADUTS gave a larger class of curves for which the algorithm is available. In this report the successive improvements of the decoding are exposed, beginning with JUSTESSEN et al. ideas which are in fact a generalisation of the decoding for REED-SOLOMON Codes.

SOMMAIRE

A-DEFINITION D'UN AG-CODE:

Dans la suite nous emploierons les notations suivantes :

p nombre premier, e un entier; $q=p^e$; K=GF(q); $\bar{K}$ une clôture algébrique de K;

X une courbe projective lisse, définie et absolument irréductible sur K (c'est à dire irréductible sur $\bar{K}$); g le genre de X;

X(K) l'ensemble des points rationnels de X sur K;

K(X) le corps des fonctions rationnelles sur X; si f est un élément de K(X) on notera (f) le diviseur qu'elle définit sur X; un tel diviseur (f) est dit principal. Un diviseur rationnel sur la courbe est de la forme : $D= \sum_{i=1}^{n} n_i p_i$ où les p_i sont des places rationnelles de X et les n_i des entiers presque tous nuls; un diviseur est dit **effectif** si tous les n_i sont positifs; le degré du diviseur D est égal à $D= \sum_{i=1}^{n} n_i$;

si D est un diviseur sur X on note:

$$L(D)=\{f\in K(X) / (f) \geq -D\}\cup\{0\};$$

L(D) est un espace vectoriel sur K dont la dimension l(D) est donnée par le théorème de RIEMANN-ROCH;

$\Omega(X)$ l'espace vectoriel des formes différentielles définies sur X; on peut définir le diviseur (ω) associé à un élément ω de $\Omega(X)$. Si D est un diviseur sur X on pose: $\Omega(D)=\{\omega\in\Omega(X) / (\omega)\geq D\}\cup\{0\}$: c'est un espace vectoriel isomorphe à $L(\mathbb{K}-D)$ où $\mathbb{K}$ est la classe canonique;

si C est une courbe plane définie sur K, son équation est : $\varphi(x,y,z)=0$ où φ est un polynôme homogène de degré m (φ est une **forme**);

enfin rappelons le théorème de RIEMANN-ROCH dans la forme où nous l'utiliserons le plus souvent par la suite:

THEOREME DE RIEMANN-ROCH:

Soit D un diviseur de degré a sur une courbe projective lisse X de genre g; on a: $\mathbf{l(D)=a-g+1+l(\mathbb{K}-D)}$

où $\mathbb{K}$ est la classe canonique de X; si le diviseur D est **non spécial**, c'est à dire **si: a>2g-2, alors: l(D)=a-g+1.**

Le code $C_L(X,p,D)$:

Soit X une courbe projective lisse, définie et absolument irréductible sur K de genre g; on considère deux diviseurs **p** et D tels que:

$\mathbf{p}=P_1+...+P_n$ où les P_i sont n points distincts de X(K);

D un diviseur effectif de X de degré a tel que: $\mathbf{2g-2<a<n}$;

les supports de **p** et D sont disjoints;

soit l'application : $ev_p : L(D) \rightarrow K^n$, $ev_p(f)=(f(P_1),....., f(P_n))$;

alors : $C_L=Im(ev_p)$.

Les paramètres de C_L sont :

* $n(C_L)= n$
* $k(C_L)=l(D)= a-g+1$ **d'aprés le théorème de RIEMANN-ROCH**
* $d(C_L) \geq n-a$.

Le code $C_\Omega(X,p,D)$:

Soit X une courbe projective lisse, définie et absolument irréductible sur K de genre g;

$\mathbf{p}=P_1+...+P_n$ où les P_i sont n points distincts de X(K);

D un diviseur rationnel effectif de X de degré a tel que: $\mathbf{2g-2<a\leq n+2g-2}$; D est donc non spécial;

les supports de **p** et D sont disjoints;

soit l'application: $res_p: \Omega(p-D) \rightarrow K^n$, $res_p(\omega)=(res(P_1,\omega),....., res(P_n,\omega))$;

alors : $C_\Omega=Im(res_p)$.

Les paramètres de C_Ω sont :

* $n(C_\Omega) = n$
* $k(C_\Omega) = n-l(D)= n-a+g-1$
* $d(C_\Omega) \geq d^* = a-2g+2$; **d^* est la distance prévue de C_Ω.**

Remarquons tout de suite que si g=0, le code C_Ω est MDS; en effet, en vertu de la borne de Singleton valable pour tout code C: $d_C \leq n_C-k_C+1$, on a nécessairement: $d(C_\Omega)=d^*=a+2=n(C_\Omega)-k(C_\Omega)+1$. Si C_Ω est un code elliptique (g=1), la borne de Singleton implique $d(C_\Omega= d^*$ ou $d^* +1$; dans le deuxième cas le code est MDS. Toutefois Katsman et Tsfasman [K-T] ont montré que "le plus souvent" la distance du code C_Ω était égale à la distance prévue.

Si le diviseur D est de degré a tel que: $2g-2<a<n$, le code C_Ω est le dual de C_L. Il s'ensuit que **la matrice génératrice** $\underline{G}$ de C_L est la matrice de contrôle de C_Ω. La matrice $\underline{G}$ est obtenue de la façon suivante :

on construit une base de l'espace L(D) (cf. par exemple [LB-R]), $\{f_s,...,f_{s-1}\}$, où s=l(D); alors $\underline{G}=||f_i(P_j)||$.

Nous donnerons dans la suite l'algorithme de décodage du code C_Ω.

B-CODES DE REED-SOLOMON:

1-DEFINITION:

Soient $q=p^e$, $n=q-1$,$d<n$; α une racine primitive nième de 1 dans K=GF(q).

Le code de REED-SOLOMON C est le code cyclique engendré par le polynôme : $g(x)=\prod_{i=1}^{d-1}(x-\alpha^i)$; à un mot $\underline{c}$ de C on associe le polynôme : $c(x)=c_0+c_1x+....+c_{n-1}x^{n-1}$, si $\underline{c}=(c_0,c_1,....+c_{n-1})$. Les paramètres du code sont [n,k=n-d+1,d] et le code est MDS. La matrice de contrôle du code C est:

$$H=\begin{pmatrix} 1 & \alpha & \alpha^2 \ldots & \alpha^{(n-1)} \\ 1 & \alpha^2 & \alpha^4 \ldots & \alpha^{2(n-1)} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ 1 & \alpha^{(d-1)} & \alpha^{2(d-1)} \ldots & \alpha^{(d-1)(n-1)} \end{pmatrix}$$

2-LES RS-CODES SONT DES AG-CODES:

Les notations étant celles de **1-** , le RS-code C est un AG-code associé à:

-la courbe $X=\mathbf{P}^1(K)$: c'est à dire la droite projective définie sur K; c'est donc une courbe lisse de genre g=0;

-le diviseur $\mathbf{p}=P_1+...+P_n$, où P_i est le point de coordonnées $(\alpha^{i-1},1)$;

-le diviseur $D=(d-2)P_\infty$, où P_∞ est le point de coordonnées (1,0).

Les paramètres du AG-code C_Ω associé à X,**p**,D sont: [n,n-d+1,d]. Une base de L(D) est: $\{1,x,x^2,...,x^{d-2}\}$; la matrice de contrôle $\underline{H}$ de C_Ω

s'obtient en divisant toutes les lignes de la matrice H précédente par sa première ligne.

3-ALGORITHME DE DECODAGE DES RS-CODES [Pt]:

Supposons que le nombre maximum t d'erreurs commises lors de la transmission d'un mot du code C soit tel que: $t \leq \lfloor (d^*-1)/2 \rfloor$ ($\lfloor\rfloor$ désignant la partie entière).

Si $\underline{w}=(w_1,...,w_n)$ est un mot reçu on a: $\underline{w}=\underline{c}+\underline{e}$ où $\underline{c} \in C$ et $wt(\underline{e}) \leq t$; les étapes du décodage sont les suivantes:

étape 1: calcul du syndrôme:

$$S= \underline{w}\,\underline{H}^T= \underline{e}\,\underline{H}^T=(S_1,...,S_{d-1});$$

étape 2: calcul du polynôme localisateur d'erreurs:

soit $\underline{\sigma}=(\sigma_t,\sigma_{t-1},....,\sigma_1,1)$ et $\underline{S}$ la matrice :

$$\underline{S} = \begin{pmatrix} S_1 & \dots & S_{t+1} \\ S_2 & \dots & S_{t+2} \\ \dots & \dots & \dots \\ S_t & \dots & S_{2t} \end{pmatrix}$$

on considère le système de t équations:

$$(A) : \underline{S}\ \underline{\sigma}^T=0 ;$$

soit u le nombre maximum d'équations linéairement indépendantes de (A): u est le nombre de positions d'erreurs commises lors de la transmission. On pose $\sigma_{u+1}=....= \sigma_t=0$, et on cherche $\underline{\sigma}=(\sigma_u,\sigma_{u-1},....,\sigma_1,1)$ solution des u premières équations de (A). Le polynôme localisateur d'erreurs est:

$$\sigma(x)=(-1)^u\sigma_u +....- \sigma_1 x^{u-1}+x^u.$$

Remarque: on connait le nombre exact d'erreurs commises ce qui ne sera pas le cas pour le décodage des A-G-codes dans le cas général.

étape 3: recherche des positions des erreurs:

l'ensemble $I(e)=\{i,0\leq i\leq n-1 \ / \ \sigma(\alpha^i)=0\}$ est l'ensemble des positions d'erreurs, et on a: card I(e)=u. Les positions des erreurs se trouvent donc en cherchant les zéros de σ dans k.

étape 4: valeurs des erreurs:

les erreurs e_j , pour $j\in I(e)$ sont solutions du système d'équations:

$$(B) \quad S_i = \sum_{j\in I(e)} z_j(\alpha^j)^i \quad , \text{ pour } i=1,...,u.$$

C-DECODAGE DES CODES DE JUSTESEN ET AL.:

Les codes géométriques introduits dans l'article [Ju] sont un peu différents des AG-codes classiques.

1-DEFINITION:

X est une courbe plane lisse définie sur K, F=0 est l'équation de X, où F est une forme de degré m. Le genre de X est donc égal à:

g= (m-1)m-2)/2;

$\mathbf{p}=P_1+...+P_n$,où les P_i sont des points distincts de X(K): $P_i=(1,y_i,z_i)$;

j est un entier tel que: **n>mj et j≥m-2** **(J);**

$V_j=\{f\in k[x,y,z] \ / \ f \text{ forme de degré } j\} \cup \{0\}$;

$val_{\mathbf{p}} : V_j \rightarrow k^n$, $val_{\mathbf{p}}(f)=(f(P_1),....., f(P_n))$.

Alors on pose: $C_l(j)=Im \ val_{\mathbf{p}}$; le code $C_l(j)$ ainsi défini a les paramètres suivants:

* $n(C_l)=n$

* $k(C_l)=\binom{j+2}{2} - \binom{j-m+2}{2} = mj-g+1$

* $d(C_l)\geq n-mj$.

On considère le code dual C(j) de $C_l(j)$. Les paramètres de C=C(j) sont:

* n(C)=n
* k(C)=n-mj+g-1
* d (C)≥ **d*=mj-2g+2** ; d* est la distance prévue.

Remarque 1: les codes $C_l(j)$ sont des AG-codes C_L associés à la courbe X, le diviseur **p** et un diviseur D ainsi défini:on choisit un élément f_0 de V_j tel que: $f_0(P_i)\neq 0$ pour tout i=1,..,n. On pose: D=(f_0). On a: degré D=mj=a; les supports de D et **p** sont bien disjoints. Précisons que le diviseur associé à une forme est effectif.

Il est alors facile de montrer que les codes $C_l(j)$ et $C_L(X,\mathbf{p},D)$ sont isomorphes. Les conditions (J) sont alors équivalentes à:

$n>mj \Leftrightarrow n>\deg D$

$j \geq m+2 \Leftrightarrow \deg D > 2g-2$

les AG-codes $C_L(X,\mathbf{p},D)$ et $C_\Omega(X,\mathbf{p},D)$ sont duaux l'un de l'autre, donc le code C défini précédemment est un AG-code de type C_Ω; la deuxième condition signifie que le diviseur D est non spécial. Les paramètres de C que nous avons donnés sont un cas particulier des paramètres d'un code C_Ω.

Dans la suite nous poserons: **k(j)=mj-g+1 pour tout entier j>0**; dans le cas où: j≥m-2, l'entier k(j) est égal à l(D) pour un diviseur effectif non spécial D de X: D=(f) où f∈V_j.

Comme l'application val(**p**) utilise la valeur des formes f de V(j) en des points P_i ayant 1 pour première coordonnée , on considère les monômes en les indéterminées y et z, de degré inférieur ou égal à j :

$f_{i,k}(y,z)=y^i z^k$, pour $i+k\leq j$;

on ordonne ces monômes de la façon suivante:

$f_{i,k} < f_{i',k'} \Leftrightarrow i+k<i'+k'$ ou [$i+k=i'+k'$ et $i>i'$];

on a donc: $f_0=1, f_1=y, f_2=z, f_3=y^2, f_4=yz, f_5=z^2, \ldots\ldots, f_{s-1}=z^j$ où $s=\binom{j+2}{2}$

On pose alors: $l=\tau(i,k)$ où: $f_l=f_{i,k}$ pour l=0,...,s-1.

La matrice de contrôle du code C est:

$\underline{H}=\|f_i(P_k)\|$ où $0\leq i\leq(s-1)$ et $1\leq k\leq n$.

JUSTESEN et Al. donne un algorithme de décodage pour le code C:

THEOREME 1: Si l'entier j vérifie (J) et si la condition suivante est vraie:

$$(J_1): \begin{cases} \text{il existe un entier } h>0 \text{ tel que :} \\ t+1 \leq k(h) > d^*-g-t \\ \text{et} \quad m-2 \leq h \leq j-m+2 \end{cases}$$

alors il existe un algorithme de décodage de C qui corrige t erreurs.

Remarque 2: la première inégalité de (J_1) implique: $\mathbf{2t+1 \leq d^*-g}$; on peut donc corriger jusqu'à $\lfloor (d^*-g-1)/2 \rfloor$ erreurs,mais on n'obtient pas en général un décodage "maximum", c'est à dire jusqu'à $\lfloor (d^*-1)/2 \rfloor$ erreurs. La deuxième inégalité implique qu'on peut associer à h et à (j-h) des diviseurs effectifs non spéciaux, comme on l'a vu dans la remarque 1.

Exposons maintenant l'algorithme de décodage du code C:

2-ALGORITHME:

On suppose que les conditions (J) et (J_1) sont vérifiées. Si $\underline{w}$ est un mot reçu on a: $\underline{w}=\underline{c}+\underline{e}$ où $\underline{c} \in C$ et $wt(\underline{e}) \leq t$; les étapes du décodage sont :

étape 1: calcul du syndrôme:

$$S= \underline{w}\, \underline{H}^T=(S_0,..,S_{s-1});$$

étape 2: calcul du polynôme localisateur d'erreurs:

on pose: $\underline{h}=\binom{h+2}{2}$= nombre de monômes de degré au plus h;

$h'=j-h$ et $\underline{h'}=\binom{h'+2}{2}$= nombre de monômes de degré au plus h';

si $i= \tau(l,r)$, on pose: $S_{l,r}=S_i$ et on considère la matrice:

$\underline{S}=\|a_{l,r}\|$ pour $l=0,...,\underline{h'}-1$, $r=0,...,\underline{h}-1$

avec: $a_{l,r}=S_{l_1+r_1,l_2+r_2}$ où: $\tau(l_1,l_2)=l$ et $\tau(r_1,r_2)=r$;

on cherche $\underline{\sigma}=(\sigma_0,\sigma_1,....,\sigma_{\underline{h}-1})$ solution du système à $\underline{h}'$ équations:

$(A_J) : \underline{S}\ \underline{\sigma}^T=0$;

on choisit $\underline{\sigma}$ de telle sorte que si on pose:

$\sigma_{ab}= \sigma_{\tau(a,b)}$ pour $\tau(a,b)\in \{0,...,\underline{h}-1\}$, le polynôme:

$\sigma(y,z)=\Sigma\ \sigma_{ab}\ y^a z^b$ soit de degré minimum en z et non divisible par F(1,y,z), F étant l'équation de la courbe X; remarquons que le degré de σ est inférieur ou égal à h.

étape 3: recherche des positions d'erreurs possibles:

l'ensemble $I(e)=\{r,1\leq r\leq n\ /\ \sigma(y_r,z_r)=0\}$ est l'ensemble des positions d'erreurs possibles; les positions d'erreurs possibles correspondent aux points $P_r=(1,y_r,z_r)$, pour r décrivant I(e). Soit: u=card I(e); on a bien sûr: $u\geq t$.

étape 4: valeurs des erreurs:

les erreurs e_r , pour $r\in I(e)$ sont solutions du système d'équations:

$$(B_J)\quad S_l = \sum_{r\in I(e)} e_r\, y_r^{\,i}\, z_r^{\,k}\quad \text{, pour tout } l=\tau(i,k)\ ,\ l=0,...,s-1.$$

3-EXEMPLE:

Nous allons donner un exemple tiré de l'article de Justesen et AL. dans lequel on peut en utilisant des propriétés géométriques améliorer la capacité de l'algorithme jusqu'à obtenir un décodage maximum. Soient:

K=GF(8)

X est la quartique de KLEIN d'équation: $F(x,y,z)=x^3y+y^3z+z^3x=0$; c'est une courbe plane lisse de degré m=4 et donc de genre g=3. La courbe X a 24 points rationnels sur k: (1,0,0);(0,1,0);(0,0,1), et 21 points de la forme:

$(1,\beta,\gamma)$, où β décrit K* et γ est une des trois racines de: $z^3+\beta^3z+\beta=0$.

$\mathbf{p}=P_1+...P_{22}$ où les P_i sont les 22 points de X(K) dont la première coordonnée est égale à 1.

j=3 ; j vérifie bien la condition (J).

On obtient un code C(3) dont les paramètres sont [22,12,8]; dans ce cas la distance du code est égale à la distance prévue. Un décodage

maximum doit décoder 3 erreurs or la capacité de l'algorithme précédent est:

$t \leq \lfloor (d^*-g-1)/2 \rfloor = 2$ et la condition (J_1) n'est satisfaite pour aucun nombre h. Pourtant on peut corriger 3 erreurs.

On considére la base de V(3) formée des 10 monômes de degré inférieur ou égal à 3:

1	y	z	y^2	yz	z^2	y^3	y^2z	yz^2	z^3	
(0,0)	(1,0)	(0,1)	(2,0)	(1,1)	(0,2)	(3,0)	(2,1)	(1,2)	(0,3)	**:(i,k)**
0	1	2	3	4	5	6	7	8	9	**:l=τ(i,k)**

Soit: h=2 donc: $\underline{h}$=6 et: h'=j-h=1 donc: $\underline{h'}$=3. Existe-t-il un polynôme localisateur d'erreurs σ:

$\sigma(y,z)=\Sigma\, \sigma_{ab}y^az^b$ pour $\tau(a,b)=0,...,(\underline{h}-1)=5$?

Supposons que 3 erreurs soient commises:

si les points correspondants aux 3 erreurs sont alignés sur la droite d'équation:

$$F_0(1,y,z)= \sigma_{10}y + \sigma_{01}z + \sigma_{00}=0 ;$$

on prend comme polynôme localisateur d'erreurs:

$$\sigma(y,z)= F_0(1,y,z);$$

et on cherche une solution: $\underline{\sigma}=(\sigma_{00},\sigma_{10},\sigma_{01},0,0,0)$ du système:$\underline{S}\ \underline{\sigma}^T=0$ qui s'écrit:

$$\begin{bmatrix} S_{00} & S_{10} & S_{01} \\ S_{10} & S_{20} & S_{11} \\ S_{01} & S_{11} & S_{02} \end{bmatrix} \begin{bmatrix} \sigma_{00} \\ \sigma_{10} \\ \sigma_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

si les points ne sont pas alignés, il existe une conique qui passe par les trois points et d'équation affine :

$$F_0(1,y,z)=y^2+\sigma_{10}y + \sigma_{01}z + \sigma_{00}=0 ;$$

on peut donc trouver un polynôme localisateur d'erreurs σ tel que:

$$\sigma(y,z)= F_0(1,y,z);$$

on cherche une solution: $\underline{\sigma}=(\sigma_{00},\sigma_{10},\sigma_{01},1,0,0)$ du système:$\underline{S}\,\underline{\sigma}^T=0$:

$$\begin{bmatrix} S_{00} S_{10} S_{01} S_{20} S_{11} S_{02} \\ S_{10} S_{20} S_{11} S_{30} S_{21} S_{12} \\ S_{01} S_{11} S_{02} S_{21} S_{12} S_{03} \end{bmatrix} \begin{bmatrix} \sigma_{00} \\ \sigma_{10} \\ \sigma_{01} \\ 1 \\ 0 \\ 0 \end{bmatrix} = 0$$

ce système se ramène au suivant:

$$\begin{bmatrix} S_{00} S_{10} S_{01} \\ S_{10} S_{20} S_{11} \\ S_{01} S_{11} S_{02} \end{bmatrix} \begin{bmatrix} \sigma_{00} \\ \sigma_{10} \\ \sigma_{01} \end{bmatrix} = \begin{bmatrix} S_{20} \\ S_{30} \\ S_{21} \end{bmatrix}$$

si α désigne une racine de: $x^3+x+1=0$ et si on a par exemple:

$S_0=S_{00}=\alpha^5$; $S_1=S_{10}=\alpha$; $S_2=S_{01}=\alpha^6$; $S_3=S_{20}=\alpha^6$; $S_4=S_{11}=\alpha$; $S_5=S_{02}=\alpha^5$; $S_6=S_{30}=\alpha^5$; $S_7=S_{21}=0$; $S_8=S_{12}=\alpha^6$; $S_9=S_{03}=\alpha^2$;

le polynôme évaluateur d'erreurs est égal à: $\sigma(y,z)= y^2+ \alpha^2 y + \alpha^5 z$; les positions d'erreurs possibles correspondent aux 5 points :

$(1,0,0);(1,1,\alpha);(1,\alpha,1);(1,\alpha^3,\alpha^3);(1,\alpha^4,1);$

on remplace les coordonnées de ces points dans le système (B_J) et on le résoud: on trouve une solution unique: $e_1=1$; $e_2=\alpha^2$; $e_3=0$; $e_4=\alpha$; $e_5=0$.

Dans tous les exemples traités, les auteurs de [Ju] adaptent l'algorithme de façon à corriger $\lfloor(d^*-1)/2\rfloor$ erreurs mais ils ne peuvent pas justifier cette capacité de correction de façon théorique.

4-REMARQUE SUR L'ALGORITHME DE [Ju]:

Les auteurs ne considèrent que des courbes planes lisses et ne conservent que les points rationnels ayant la première coordonnée égale à 1. Cette situation a été généralisée par [S-V] à des courbes projectives et à

des diviseurs quelconques; toutefois on retrouve dans le travail de [S-V] la même limitation sur la capacité de l'algorithme.

Dans la suite nous appellerons **décodage "maximum"** un décodage qui corrige $\lfloor (d^*-1)/2 \rfloor$ erreurs où d^* est la distance prévue du code.

D-DECODAGE DES AG-CODES [S-V]:

1-DEFINITION:

On se place dans la situation d'un AG-code C_Ω associé à :

-une courbe projective lisse X définie et absolument irréductible sur K, de genre g,

- un diviseur $\mathbf{p} = P_1 + \ldots + P_n$, P_i n points distincts de X(K)
- un diviseur effectif D de degré a tel que : $2g-2<a\leq n+g-1$
- les supports de **p** et D sont disjoints.

} (A-G)

Rappelons enfin que les paramètres de C_Ω sont :

* $n(C_\Omega) = n$
* $k(C_\Omega) = n-a+g-1$
* $d(C_\Omega) \geq d^* = a-2g+2$; d^* est la distance prévue de C_Ω.

La matrice de contrôle de C_Ω est: $\underline{H}=\|f_i(P_j)\|$ où $\{f_0,\ldots,f_{s-1}\}$ est une base de L(D): $s=l(D)=a-g+1$. Les auteurs prouvent le résultat suivant:

THEOREME 3 [S-V] : Si l'entier a vérifie la condition **(A-G)** et si la condition suivante est vérifiée:

$(S\text{-}V_1)$ { **il existe un diviseur rationnel effectif F de X tel que :**
deg F = b ; supp F $\subset$ supp D
l(F) > t
a-b > t+2g-2

alors il existe un algorithme de décodage de C_Ω qui corrige t erreurs.

Remarque 3: on peut réécrire les inégalités de **(S-V$_1$)** sous la forme:

$$b-g+1(\mathbb{K}-F) \geq t$$
$$d^* - b \geq t+1$$

on a donc: **$2t+1 \leq d^*-g+1(\mathbb{K}-F)$** et si F est non spécial (ce qui n'est pas exigé) : **$2t+1 \leq d^*-g$** ; l'inégalité: $a-b > t+2g-2$ implique par contre que le diviseur D-F doit être non spécial.

Le décodage du code est la généralisation du décodage de [Ju]; avant de l'exposer, faisons la remarque suivante:

soit $\underline{w}$ un élément de K^n; si f est un élément de L(D) on définit le **syndrôme de $\underline{w}$ associé à f :** $S(\underline{w},f)= \sum_{j=1}^{n} w_j f(P_j)$; on a alors bien sûr:

$\underline{w} \in C_\Omega$ si et seulement si : $S_i=S(\underline{w},f_i)=0$ pour tout i=0,..,(s-1);

d'autre part, si on note $\{g_0,..,g_{l-1}\}$ une base de L(F) et $\{h_0,...,h_{r-1}\}$ une base de L(D-F) alors h_ig_j est un élément de L(D) pour tous i et j; on posera alors: $S_{ij}(\underline{w}) =S(\underline{w}, h_ig_j)$.

L'algorithme de décodage du code C_Ω est le suivant:

2-ALGORITHME:

On suppose que les conditions (A-G) et (S-V$_1$) sont vérifiées. Si $\underline{w}$ est un mot reçu on a: $\underline{w}=\underline{c}+\underline{e}$ où $\underline{c} \in C_\Omega$ et $wt(\underline{e}) \leq t$; les étapes du décodage sont :

étape 1: calcul du syndrôme:

$S= (S_0,..,S_{s-1})$ où $S_i=S(\underline{w},f_i)$;

étape 2: calcul de la fonction localisatrice d'erreurs:

On cherche $\underline{\sigma}=(\sigma_0,\sigma_1,....,\sigma_{l-1})$ solution non triviale du système :

$$(A) : \sum_{j=0}^{l-1} S_{ij}(\underline{w})\,\sigma_j = 0 \quad \text{pour } i=0,..,(r-1);$$

on peut bien sûr écrire ce système sous forme matricielle en utilisant les matrices: $\underline{H'}=\|g_i(P_j)\|$ de type lxn et $\underline{H''}=\|h_i(P_j)\|$ de type rxn.La fonction localisatrice d'erreurs est l'élément g_σ de L(F):

$$g_\sigma = \sum_{i=0}^{l-1} \sigma_i\, g_i$$

étape 3: recherche des positions d'erreurs possibles:

l'ensemble $I(e)=\{k, 1\le k\le n \ / \ g_\sigma(P_k)=0\}$ est l'ensemble des positions d'erreurs possibles; les erreurs possibles correspondent aux points P_k pour k décrivant I(e). Soit: u =card I(e); on a bien sûr: $u\ge t$.

étape 4: valeurs des erreurs:

les erreurs e_k , pour $k\in I(e)$ sont solutions du système d'équations:

$$(B) \quad S_l = \sum_{k\in I(e)} e_k\, f_l(P_k) \quad \text{, pour tout l , } l=0,...,s-1.$$

la complexité de l'algorithme est : $O(n^3)$.

3-CAPACITE DE L'ALGORITHME:

Soit: $t(C_\Omega)= \mathrm{Max}_F \ \mathrm{Inf}\ \{l(F), a-b-2g+2\} -1$, où le Max est pris sur tous les diviseurs effectifs F de degré b dont le support est inclus dans le support de D et qui vérifie les conditions (A-G); alors l'algorithme de décodage décode t erreurs pour tout t tel que: $t\le t(C_\Omega)$

cas particulier: soit D=aQ où Q est un point de X(K) n'appartenant pas au support de **p** (i.e. distinct des P_i , i=1,..,n); on a alors:

$t(C_\Omega) \geq \lfloor (a-3g+1)/2 \rfloor = \lfloor (d^*-g-1)/2 \rfloor$;

si donc: g=0 ou g=1 et d* est impair on a: $t(C_\Omega) = \lfloor (d^*-1)/2 \rfloor$.

Pour un tel diviseur D, V. Yu. Kratchkovskii [K] donne une amélioration de l'algorithme que nous allons maintenant exposer.

4-LE CAS PARTICULIER: D=aQ , [K] :

4-1-EXISTENCE DE L'ALGORITHME:

On se propose de décoder le code C_Ω défini comme dans le §1 précédent . L'amélioration viendra de ce qu'il est possible, par la forme particulière du diviseur D, de définir un ordre sur les éléments d'une base de L(D) généralisant l'ordre canonique de la base polynomiale des codes de R-S.

On notera comme précédemment:

$s=l(D)=a-g+1$
$d^* = a-2g+2$ la distance prévue;

si f est un élément de L(D)=L(aQ), f a un unique pôle en Q et on notera $\nu_Q(f)$ l'ordre (positif) de f en ce pôle. On construit une base de L(D): $\{f_0,...,f_{s-1}\}$ par récurrence et de telle sorte que:

$$\nu_Q(f_0) < \nu_Q(f_1) < ... < \nu_Q(f_{s-1});$$

si s(i) est l'ordre du (i+1)-ième élément f_i de la base , on a :

$$f_i \in L(s(i)Q) \text{ et } l(s(i)Q)=i+1;$$

d'où d'aprés le théorème de Riemann-Roch: $s(i)=i+g-l(\mathbb{K}-s(i)Q)$;

$\underline{H}=\| f_i(P_j) \|$ est la matrice de contrôle du code C_Ω .

THEOREME 3: Si l'entier a vérifie la condition **(A-G)** et si la condition suivante est vérifiée:

$$(\mathbf{K}_1)\begin{cases}\textbf{il existe deux entiers m et m' tels que}\\ \mathbf{m \geq t}\\ \mathbf{m + m' \leq d^*\text{-}1}\\ \mathbf{m' \geq t + g}\end{cases}$$

alors il existe un algorithme de décodage de C_Ω qui corrige t erreurs.

Remarque 4: les inégalités (K_1) donne la capacité de correction suivante: **$2t+1 \leq d^*-g$**. Montrons que les conditions (K_1) sont équivalentes aux conditions $(S\text{-}V_1)$. Supposons m et m' vérifiant (K_1) et posons: s(m)=b et F=bQ; alors l(F)=m+1; par suite: $m \geq t$ équivaut à l(F)>t.

De plus: b=m+g-l($\mathbb{K}$-s(m)Q) implique: $a-b \geq a-m-g$;
la condition: $m+m' \leq d^*-1$ peut s'écrire: $m+m' \leq a-2g+1$;
d'où: $a-m \geq 2g-1+m'$ et comme: $m' \geq t+g$, on a:
$a-m \geq 3g-1+t$; par suite: $a-b \geq t+2g-1$ ou encore: $a-b > t+2g-2$.

Exposons maintenant l'algorithme de décodage du code C_Ω:

4-2-ALGORITHME:

On suppose que les conditions (A-G) et (K_1) sont vérifiées. Si $\underline{w}$ est un mot reçu on a: $\underline{w}=\underline{c}+\underline{e}$ où $\underline{c} \in C_\Omega$ et $wt(\underline{e}) \leq t$; les étapes du décodage sont:

étape 1: calcul du syndrôme:

$$S = \underline{w}\,\underline{H}^T = (S_0,..,S_{s-1});$$

étape 2: calcul de la fonction localisatrice d'erreurs:

on pose: $S_{ij} = \sum_{k=1}^{n} w_k f_i(P_k) f_j(P_k)$ et on considère la matrice :

$\underline{S}=||S_{ij}(\underline{w})||$ pour i=0,..,m'-1 et j=0,..,m ;

cette matrice peut être calculée de la façon suivante:

soient les matrices: $\underline{H'}=||f_i(P_k)||$ pour i=0,..,m'-1 et k=1,..,

et $\underline{H''}=||f_j(P_k)||$ pour j=0,..,m et k=1,..,n

et la matrice diagonale:

$$\text{diag}\,(\underline{w}) = \begin{bmatrix} w_1 & 0 & 0 & \ldots\ldots & 0 \\ 0 & w_2 & 0 & \ldots\ldots & 0 \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ 0 & 0 & 0 & \ldots\ldots & w_n \end{bmatrix}$$

alors: $\underline{S}= H'\ \text{diag}(\underline{w})\ H''^{T}$.

On cherche $\sigma=(\sigma_0,\sigma_1,\ldots,\sigma_m)$ solution non triviale du système :

(A) : $\underline{S}\ \sigma^T=0$;

F étant le diviseur défini dans la remarque 4, la fonction localisatrice d'erreurs est l'élément g_σ de L(F):

$$g_\sigma = \sum_{i=0}^{m} \sigma_i f_i ;$$

étape 3: recherche des positions d'erreurs possibles:

l'ensemble $I(e)=\{k,1\leq k\leq n\ /\ g_\sigma(P_k)=0\}$ est l'ensemble des positions d'erreurs possibles; les erreurs possibles correspondent aux points P_k pour k décrivant I(e). Soit: u=card I(e)≥t et posons $I(e)=\{k_1,k_2,\ldots,k_u\}$.

étape 4: valeurs des erreurs:

Considérons la matrice: $\underline{E_u}=||\lambda_{ij}||$ pour i=1,..,u et j=1,..,n et où $\lambda_{ij}=1$ si $j=k_i$ et 0 sinon. On en déduit la matrice $\underline{H_u} = \underline{H}\ \underline{E_u}^T$ dont les colonnes sont celles de $\underline{H}$ qui correspondent aux positions des erreurs. On cherche une solution $\underline{z}=(z_1,..,z_u)$ du système:

(B) $S = \underline{H_u}\ \underline{z}^T$;

alors le vecteur erreur est: $\underline{e} = \underline{z}\,\underline{E_u}$.

la complexité de l'algorithme est : $O(t^3)$.

4-3-CAPACITE DE L'ALGORITHME:

On a vu que, si les conditions (K_1) étaient remplies, le nombre maximum t d'erreurs corrigées vérifiait: $2t+1 \leq d^*-g$ ou encore:

$$2t+g \leq d^*-1 = s-g.$$

On ne peut donc pas avoir un décodage "maximum"; toutefois, si $s \geq 3g$, on obtient la correction "maximum" dans l'étape 4. En effet la fonction localisatrice d'erreurs g_σ est un élément de $L(s(m)Q)$; elle a donc un pôle en Q d'ordre au plus s(m) donc au plus s(m) zéros avec:

$$s(m) = m+g-l(\mathbb{K}-s(m)Q) \leq m+g.$$

Il y a donc au plus (m+g)-positions d'erreurs. Les u erreurs où $u \leq m+g$, seront corrigées si: $rang(\underline{H_u})=u$ ce qui est équivalent à: $u \leq s-g$; une condition suffisante est donc: $m+g \leq s-g$ soit encore: $m \leq s-2g$. En prenant m=t on a: $2t \leq 2s-4g$ et comme $d^*-1=s-g$ on en déduit:

$$2t- d^* +1 \leq s-3g;$$

si donc: $s \geq 3g$ on pourra décoder, dans l'étape 4 , un nombre d' erreurs t tel que: $2t+1 = d^*$.

Remarque 5: Si on choisit une solution σ de (A) telle que le polynôme:
$\sigma_0+\sigma_1 z+...+\sigma_m z^m$ soit de degré minimum, on restreind le nombre de zéros de g_σ.

E-DECODAGE "MAXIMUM" :

Les conditions $(S\text{-}V_1)$: $l(F) > t$ et $a-b > t+2g-2$ implique:

$$2t \leq d^* -1-g+l(\mathbb{K}-F).$$

Une première façon d'améliorer la capacité de décodage de l'algorithme est donc de trouver un diviseur F pour lequel $l(\mathbb{K}-F)$ n'est pas nul (et au mieux égal à g); c'est ce qui est fait dans l'article [S-V].

Une deuxième façon, et c'est ce que fait Pellikaan [Pl], consiste à appliquer l'algorithme plusieurs fois pour différents diviseurs F_i bien choisis; c'est ce qui est montré dans [Pl] pour des courbes maximales et dans [V] dans un cas plus général. Il est prouvé qu'on peut décoder jusqu'à $\lfloor (d^*-1)/2 \rfloor$-erreurs mais l'algorithme n'est pas effectif sauf si g=1. Pour exposer les résultats de ces articles nous avons besoin de quelques définitions et notations supplémentaires.

1-DEFINITIONS :

Si G est un ensemble et l un entier, $l \geq 2$, on notera G^l le produit cartésien de l exemplaires de G;

soit X une courbe projective lisse définie et absolument irréductible sur GF(q) de genre g on pose :

$\mathbb{D}_k = \{D/D$ diviseur rationnel effectif de X, de degré k$\}$ et $\mathbf{c_k} = \# \mathbb{D}_k$;

Si on considère le polynôme: $Z(X,T) = \sum_{k \geq 0} c_k T^k$

alors **la fonction Zéta de la courbe X** est telle que: $\zeta(X,s) = Z(X,q^{-s})$;

le groupe de Picard de X , Pic(X), est le groupe abélien de tous les diviseurs de X modulo les diviseurs principaux ; si D est un diviseur, on note [D] sa classe dans Pic(X) ;

la jacobienne de X, Jac(X) est le sous groupe de Pic(X) des diviseurs de degré 0 modulo les diviseurs principaux et on posera:

h=#Jac(X) .

On a de plus:

$$Z(X,T)= P(T)\ (1-T)^{-1}\ (1-qT)^{-1}\ , \text{ où } : P(T) = \sum_{i=0}^{2g} p_i T^i$$

$$c_{g-1}=\Big(\sum_{i=g+1}^{2g} p_i - \sum_{i=0}^{g-1} p_i\Big) / (q-1)$$

$$h=P(1)= \sum_{i=0}^{2g} p_i = \prod_{i=1}^{g}(1-\alpha_i)(1-\bar{\alpha}_i)$$

où α_i et $\bar{\alpha}_i$ sont complexes conjugués de module égal à $\sqrt{q}$;

si K=GF(q) et N_q=#X(K), on a la borne de Weil :

$$|N_q-q-1|\leq 2g\sqrt{q}\ ;$$

si q est un carré, on dira qu'une courbe X est **maximale** si son nombre de points rationnels N_q sur GF(q) atteind la borne supérieure de l'inégalité de Weil:

$$N_q=q+1+2g\sqrt{q}.$$

2-EXISTENCE DE L'ALGORITHME:

THEOREME 4 [Pl]: Soit q≥3 et C_Ω(**p**,D) un A-G-code où a=deg(D) est tel que: 4g-1≤ a ≤ n+2g-2; si la condition suivante est vérifiée:

(**P**) **il existe un nombre c, c>1, indépendant de la courbe X, tel que : $h \geq c\ c_{g-1}$**

alors il existe un algorithme de complexité en temps $O(n^4)$ et de complexité en espace $O(n^3)$ qui corrige jusqu'à $t^*=\lfloor(d^*-1)/2\rfloor$-erreurs.

Le corollaire suivant donne un exemple des courbes pour laquelle la condition du théorème 4 est remplie:

THEOREME 5 [Pl]: Si $q \geq 3$ et si X est une courbe maximale la condition **(P)** est vérifiée pour $c = q-1$ et il existe un décodage "maximum".

S. G. Vladut a ensuite prouvé que la condition (P) était vraie pour une classe plus générale de courbes projectives:

THEOREME 6 [V] : La condition **(P)** est vraie pour $c=(q-1)/2$ dans les cas suivants:

* $q \geq 37$
* $q \geq 16$ et le genre de X est suffisamment grand.

Dans le paragraphe suivant nous allons donner les grandes lignes de la démonstration du théorème 4.

3- INDICATIONS SUR LA PREUVE DE TH.4:

Nous reprenons les notations et les résultats de [Pl]; la proposition 7 prouve l'existence d'une fonction localisatrice d'erreurs; nous avons repris la démonstration de Pellikaan pour montrer comment il a pu ensuite améliorer la capacité de décodage de l'algorithme.

Si V est un K-espace vectoriel, on notera V* son dual, c'est à dire l'espace vectoriel sur K des applications linéaires de V dans K.

Si $\underline{w}$ est un mot de K^n, on définit son syndrôme $S(\underline{w},f)$ pour tout élément f de L(D) comme on l'a fait dans la remarque 3. F étant un diviseur de support disjoint de $\{P_1,\ldots,P_n\}$, on définit de plus **l'application localisatrice d'erreurs $E_{\underline{w}}$** par:

$$E_{\underline{w}}: L(F) \rightarrow L(D-F)^* \ ; \ E_{\underline{w}}(f)= [\ h \rightarrow S(\underline{w},f\,h)\ ]$$

il est clair que si: $\underline{w}=\underline{c}+\underline{e}$, alors: $E_{\underline{w}}=E_{\underline{e}}$ et si $Q_1,...,Q_t$ sont les positions d'erreurs, alors $L(F - \sum_{j=1}^{t} Q_j)$ est un sous espace vectoriel de Ker $E_{\underline{w}}$. On a la proposition suivante qui reprend les conditions $(S\text{-}V_1)$:

PROPOSITION 7:

1) si $t< l(F)$ on a : $L(F - \sum_{j=1}^{t} Q_j) \neq \{0\}$

2) si $deg(D-F)> t+2g-2$ on a : Ker $E_{\underline{w}} = L(F-\Sigma_{j=1,..,t}\ Q_j)$.

preuve: 1) dire qu'un élément f de L(F) appartient en fait à $L(F- \sum_{j=1}^{t} Q_j)$, c'est dire que f a un zéro en chacun des points Q_j; ceci se traduit par t conditions linéaires indépendantes et comme $t<l(F)$,$L(F - \sum_{j=1}^{t} Q_j)$ n'est pas réduit à $\{0\}$.

2) le théorème de Riemann-Roch appliqué à L(D-F) donne:

$l(D-F)= deg(D-F)-g+1$, puisque le diviseur D-F est non spécial; on a donc: $l(D-F)> t+g-1$, d'où $l(D-F)\geq t$.

Il s'ensuit que l'application:

$$\phi(F,\underline{Q}) : L(D-F) \rightarrow K^t \qquad h \rightarrow (h(Q_1),...,h(Q_t))$$

est surjective; de plus le noyau de $\phi(F,\underline{Q})$ est égal à $L(F - \sum_{j=1}^{t} Q_j)$.

Pour tout j, j=1,..,t, soit h_j un élément de L(D-F) tel que: $\phi(F,Q)(h_j)=(0,..,1,..,0)$ (1 à la j-ième coordonnée) et posons:

$P_{i_j}= Q_j$;

alors si f est un élément de ker $E_{\underline{w}}$, on a:

$S(w,fh_j)=0=e_{i_j}f(Q_j)$ donc: $f(Q_j)=0$ pour tout j et f est un élément de $L(F - \sum_{j=1}^{t} Q_j)$; d'où 2).

Pour un mot donné $\underline{w}$ affecté de t-erreurs cette proposition montre que, s'il existe un diviseur F tel que $l(F)>t$ et $\phi(F,\underline{Q})$ soit surjective où $\underline{Q}$ est l'ensemble des positions des erreurs, alors il existe une fonction localisatrice d'erreurs f qui est un élément de $\ker E_{\underline{w}}$; les positions des erreurs se trouvent parmi les zéros de f. Le calcul des valeurs des erreurs découle de la proposition 8.

On pose: $\mathbf{t^*=\lfloor(d^*-1)/2\rfloor}$; on suppose comme précédemment que
$\{Q_1,...,Q_t\} =\{P_{i_1},...,P_{i_t}\}$ pour $t\leq t^*$

est l'ensemble des positions d'erreurs affectant le mot $\underline{w}$ et qu'il existe un élément non nul f dans $\ker E_{\underline{w}}$ tel que les points $Q_1,...,Q_t$ soient parmi les zéros de f ; soit :

$$\underline{Q}=\{Q_1,...,Q_u\} =\{P_{i_1},...,P_{i_u}\} \text{ , } u\geq t \text{ ,}$$

l'ensemble des zéros de f appartenant à $\{P_1,...,P_n\}$.

On considère les applications suivantes:

$$S: K^n \rightarrow L(D)^* \qquad \underline{w} \rightarrow S(\underline{w}): h \rightarrow S(\underline{w},h)$$

on a bien sûr: $\underline{w}\in C_\Omega \Leftrightarrow S(\underline{w})=0$

$$S_{\underline{Q}}: K^u \rightarrow L(D) \qquad \underline{v} \rightarrow S_{\underline{Q}}(\underline{v}): g \rightarrow \sum_{j=1}^{u} v_j\, g(Q_j)$$

$$i_{\underline{Q}}: K^u \rightarrow K^n \quad \text{telle que: } i_{\underline{Q}}(\underline{v})_i=v_i \text{ si } i\in \{i_1,...,i_u\} \text{ et } =0 \text{ sinon}$$

$$\pi_{\underline{Q}}: K^n \rightarrow K^u \qquad \underline{w} \rightarrow (w_{i_1},..., w_{i_u})$$

on a donc: $\pi_{\underline{Q}}\circ i_{\underline{Q}}=id_{K^u}$.

On pourra décoder le mot $\underline{w}$ en utilisant le résultat suivant:

PROPOSITION 8: On suppose que F est effectif ou que $a\leq n+2g-2$.
Si on a $\deg (D-F)>t^*+g-2$ et $g\leq t^*$, alors l'équation:

$$S_{\underline{Q}}(\underline{x})=S(\underline{w}) \qquad \text{où } \underline{x}=(x_1,...,x_u)$$

a une unique solution $\underline{x}=\pi_{\underline{Q}}(\underline{e})$.

On a donc le corollaire :

COROLLAIRE 9: Soit $C_\Omega(\mathbf{p},D)$ un AG-code de longueur n de distance prévue d* sur une courbe de genre g; on suppose de plus que:

$2g-2< a=\deg D \leq n+2g-2$ et $g \leq t^*$;

alors s'il existe un s-uple de diviseurs $\underline{F}=(F_1,..,F_s)$ tel que pour tout i, i=1,..,s :

F_i ait un support disjoint de $\{P_1,..,P_n\}$

$l(F_i)>t^*$

$\deg (D-F_i) > t^*+g-2$

et si de plus pour tout t*-uple $\underline{Q}=(Q_1,..,Q_{t^*})$ où $\{Q_1,..,Q_{t^*}\}\subset\{P_1,..,P_n\}$, il existe au moins un i, $1\leq i\leq s$, tel que $\phi(F_i,\underline{Q})$ soit surjective,alors il existe un algorithme de décodage qui corrige jusqu'à t* erreurs.

Montrons maintenant comment on peut trouver un s-uple de diviseurs $\underline{F}=(F_1,..,F_s)$ vérifiant les hypothèses du corollaire 9.

LEMME 10: a) Si D_0 est un élément de $\mathbb{D}_k$ on définit l'application:

$\psi_{0,k}: \mathbb{D}_k \rightarrow Jac(X)$ telle que: $\psi_{0,k}(D)=[D-D_0]$;

alors si: $k\geq g$, $\psi_{0,k}$ est surjective;

b) soient s,k,r trois entiers tels que: $s \geq 2$; $k \geq g$; $r \leq g-1$; considérons l'application $\psi_r^s: \mathbb{D}_r^s \rightarrow Jac(X)^{s-1}$ telle que:

$\psi_r^s(D_1,...,D_s)=([D_1-D_2],...,[D_{s-1}-D_s])$.

si ψ_r^s n'est pas surjective, alors il existe un s-uple: $\underline{F}=(F_1,...,F_s)$ dans $\mathbb{D}_k^s$ tel que $\psi_k^s(\underline{F})$ n'est pas dans l'image de ψ_r^s.

Preuve: a) c'est une conséquence du théorème de Riemann-Roch: soit J=[F] un élément quelconque de Jac(X), F est un diviseur de degré 0 représentant la classe J; comme $\deg(D_0+F)=k\geq g$, on a:

$l(D_0+F)=k-g+1\geq 1$

donc il existe un élément f non nul dans $L(D_0+F)$ tel que:

$D= (f)+D_0+F \geq 0$

et alors: $\psi_{0,k}(D)=[D-D_0] =[F] =J$.

b) se déduit facilement de a).

Ce lemme est essentiel pour démontrer la proposition suivante :

PROPOSITION 11: Soient s et a deux entiers tels que:
$s \geq 2$ et $a = \deg D \geq 4g-1$; si une des deux situations suivantes est réalisée:

1) a est impair et ψ^s_{g-1} n'est pas surjective

2) a est pair et ψ^s_{g-2} n'est pas surjective

alors il existe un s-uple $\underline{F}$ dans $\mathbb{D}^s_{g+t^*}$ tel que pour tout t*-uple de positions d'erreurs: $\underline{Q}=(Q_1,..,Q_{t^*})$, il existe au moins un i, $1 \leq i \leq s$, tel que l'application $\phi(F_i,\underline{Q})$ soit surjective.

Remarque 7: Dans [Pl], Pellikaan donne la condition $a \geq 4g-2$.

Or $t^* = \lfloor (a-2g+1)/2 \rfloor$; donc $t^* = a/2 - g$ si a est pair et $t^* = (a+1)/2 - g$ si a est impair; donc si a est pair, égal à 4g-2, la condition $t^* \geq g$ nécessaire dans la proposition 8, n'est pas satisfaite.

Preuve de la proposition: D'aprés le Lemme 10, il existe un s-uple $\underline{F}$ dans $\mathbb{D}^s_{g+t^*}$ tel que:

$\psi^s_{g+t^*}(\underline{F})$ n'est pas dans l'image de ψ^s_{g-1} si a est pair

$\psi^s_{g+t^*}(\underline{F})$ n'est pas dans l'image de ψ^s_{g-2} si a est impair.

Soit $\underline{Q}=(Q_1,..,Q_{t^*})$ un t*-uple de positions d'erreurs; supposons que pour tout i, $1 \leq i \leq s$, l'application:

$$\phi(F_i,\underline{Q}) : L(D-F_i) \rightarrow K^{t^*} \text{ ne soit pas surjective;}$$

on a: $\ker \phi(F_i,\underline{Q}) = L(D-F_i-\sum_{j=1}^{t^*} Q_j)$.

Si pose: $T_i = D-F_i-\sum_{j=1}^{t^*} Q_j$, $\phi(F_i,\underline{Q})$ non surjective est équivalent à:

$$l(D-F_i) < t^* + l(T_i) ;$$

comme $D-F_i$ est un diviseur non spécial, on en déduit que:

$$l(T_i) > a-2g-2t^*+1 ;$$

en appliquant le théorème de Riemann-Roch encore une fois, on obtient:

$$\dim \Omega(T_i) = l(T_i) - \deg T_i + g - 1 > 0 ;$$

donc pour tout i, $1 \leq i \leq s$, il existe une différentielle ω_i , telle que: $(\omega_i) \geq T_i$. Si on pose: $E_i = (\omega_i) - T_i = \mathbb{K} - T_i$, E_i est un diviseur effectif de degré (g-1) si a est impair et (g-2) si a est pair et de plus:

$$E_i - F_i = \mathbb{K} - D - \sum_{j=1}^{t^*} Q_j \text{ ne dépend pas de i;}$$

donc pour tout couple (i_1, i_2), $1 \leq i_1 < i_2 \leq s$, on a:

$$[E_{i_1} - F_{i_1}] = [E_{i_2} - F_{i_2}] \text{ d'où } [E_{i_1} - E_{i_2}] = [F_{i_1} - F_{i_2}]$$

et par suite: $\psi_{g-1}^{s}(\underline{E}) = \psi_{g+t^*}^{s}(\underline{F})$ si a est impair

$\psi_{g-2}^{s}(\underline{E}) = \psi_{g+t^*}^{s}(\underline{F})$ si a est pair

cotrairement à l'hypothèse.

Nous avons donc ramené le problème de l'existence du s-uple de diviseurs $\underline{F}$ vérifiant les hypothèses du corollaire 9, à celui du choix d'un entier s tel que ψ_{g-1}^{s} ou ψ_{g-2}^{s} ne soit pas surjective. Comme l'image de ψ_{g-2}^{s} est incluse dans l'image de ψ_{g-1}^{s}, nous allons nous intéresser seulement à l'application ψ_{g-1}^{s} .

PROPOSITION 12 : Si la courbe X est telle que il existe un nombre c, c>1 tel que: $h \geq c\, c_{g-1}$, alors on peut trouver un entier s tel que ψ_{g-1}^{s} ne soit pas surjective et on a s=O(g).

preuve: on a $h = \prod_{i=1}^{g} (1-\alpha_i)(1-\bar{\alpha}_i)$ où α_i et $\bar{\alpha}_i$ sont complexes conjugués de module égal à $\sqrt{q}$; on a donc: $h \leq (1+\sqrt{q})^{2g}$.. Alors si s est le plus petit entier tel que:

$s > 2g \log_c(1+\sqrt{q})$ on a:
$c^s > (1+\sqrt{q})^{2g} \geq h$ et comme: $h^s \geq c^s c_{g-1}^s$ on en déduit: $h^{s-1} > c_{g-1}^s$;
donc ψ_{g-1}^s n'est pas surjective et on a bien s=O(g).

L'algorithme est le suivant:

4- ALGORITHME [Pl]:

Données: courbe X de genre g ;
diviseurs $p=P_1+..+P_n$
D tel que $4g-1 \leq a=\deg D \leq n+2g-2$;
$t^*=\lfloor (a-2g+1)/2 \rfloor$;
s entier , $s \geq 2$, tel que ψ_{g-1}^s ou ψ_{g-2}^s ne soit pas surjective (suivant la parité de a);
$\underline{F}=(F_1,....,F_s)$ où pour chaque i: $\deg F_i=k=t^*+g$, le support de F_i est disjoint de $\{P_1,..,P_n\}$ et $\psi_k^s(\underline{F})$ n'est pas dans l'image de ψ_{g-1}^s (ou ψ_{g-2}^s).

Entrée: $\underline{w}$ mot reçu, $\underline{w}$ élément de K^n;

Algorithme $A(F_1,...,F_s)$:

a1- Exécuter en parallèle les subroutines $B(F_i)$ pour i=1,..,s.

a2- **a2-1**: Si $A_i=1$ pour un i , i=1,..,s : print: " le mot reçu est décodé par ",B_i .
a2-2: Si $A_i=0$ pour tout i , i=1,..,s :print: " le mot reçu a plus de t* erreurs .

a3- Fin.

Subroutine $B(F_i)$:

b1- Calculer ker $E_{\underline{w}}$; $E_{\underline{w}}$: $L(F_i) \rightarrow L(D-F_i)^*$;
b1-1: Si ker $E_{\underline{w}} = \{0\}$ alors $A_i=0$.

b1-2: Si ker $E_{\underline{w}} \neq \{0\}$, alors choisir f non nul dans ker $E_{\underline{w}}$ et soit $\underline{Q}=(Q_1,...,Q_u)$ où $Q_1,...,Q_u$ sont les zéros de f parmi $\{P_1,...,P_n\}$.

b2- Chercher une solution de $S_{\underline{Q}}(\underline{x}) = S(\underline{w})$, $x=(x_1,...,x_u)$.

b2-1: Si $S_{\underline{Q}}(\underline{x}) = S(\underline{w})$ n'a pas ou a plus d'une solution, alors $A_i=0$.

b2-2: Si $S_{\underline{Q}}(\underline{x}) = S(\underline{w})$ a une solution unique $\underline{x}_0$ calculer $wt(\underline{x}_0)$.

b2-2-1: Si $wt(\underline{x}_0)>t^*$, alors $A_i=0$.

b2-2-2: Si $wt(\underline{x}_0)\leq t^*$, alors $A_i=1$ et $B_i=\underline{w}-i_{\underline{Q}}(\underline{x}_0)$.

b3- Fin.

Le théorème suivant est une conséquence des résultats énoncés dans le paragraphe 3.

THEOREME 13:

Soit $C_\Omega(X,\mathbf{p},D)$ un AG-code de longueur n et de distance prévue d* sur une courbe X de genre g. On suppose que: $4g-1\leq a=\deg D\leq n+2g-2$ et on pose $t^*=\lfloor (d^*-1)/2 \rfloor$.

S'il existe un entier s, s>1, tel que ψ_{g-2}^{s} ne soit pas surjective dans le cas a pair et ψ_{g-1}^{s} ne soit pas surjective dans le cas a impair , alors on peut trouver un s-uple $\underline{F}=(F_1,...,F_s)$ de diviseurs de degré $g+t^*$ et de support disjoint de celui de **p** tel que l'algorithme $A(F_1,...,F_s)$ décode $C_\Omega(X,\mathbf{p},D)$ jusqu'à t* erreurs avec une complexité $O(n^3s)$ en espace et $O(n^3)$ en temps.

Le théorème 4 découle alors du théorème 13 et de la proposition 12.

5- REMARQUES SUR L'ALGORITHME [PI]:

5-1: Pour pouvoir appliquer l'algorithme il faut que les diviseurs F_i associés au s-uple $\underline{F}$, dont l'existence est assurée par la proposition 11, aient un support disjoint de $\{P_1,..,P_n\}$; ceci est toujours possible mais

alors on n'est pas assuré que les F_i soient effectifs; cette restriction n'est pas génante pour le décodage cf. proposition 8. On doit construire des bases de $L(D-F_i)$ et $L(F_i)$ où $D-F_i$ et F_i ne sont pas forcément effectifs: la construction exposée dans [LB-R] s'applique encore.

5-2: Pour une courbe elliptique X (g=1) sur K=GF(q) ayant plus d'un point rationnel la mise en oeuvre de l'algorithme est simple comme le signale Pellikaan: la jacobienne de X est isomorphe à X(K): on a h>1 et de plus $a_{g-1}=a_0=1$ donc ψ_0^2 n'est pas surjective. Il suffit donc de choisir deux diviseurs F_1 et F_2 ayant un support disjoint de $\{P_1,..,P_n\}$, de degré t*+1 et tels que: $[F_1]\neq [F_2]$ pour décoder jusqu'à t*-erreurs.

F-CONCLUSION:

Le décodage des codes géométriques jusqu'à t*-erreurs repose donc maintenant sur une connaissance approfondie des jacobiennes des courbes projectives.

Pour un AG-code donné, il faut trouver un entier s tel que l'une des applications ψ_{g-1}^s ou ψ_{g-2}^s ne soit pas surjective: si X est maximale ou si X vérifie les hypothèses du théorème 6, l'entier s est donné par la proposition 12.

Puis il faut avoir explicitement un s-uple $\underline{F}$ de diviseurs de degré g+t* dont l'existence est assurée par la proposition 11; dans le cas g=1 c'est possible, mais pour un genre plus élevé la question reste posée.

Je remercie A. Thiong-Ly pour ses commentaires avisés sur la rédaction de ce rapport.

BIBLIOGRAPHIE:

[Ju] J. JUSTESSEN, K.J. LARSEN, A. HAVEMOSE, H.E. JENSEN, T. HOHOLDT: "Construction and decoding of a class of algebraic geometry codes", IEEE Trans. Info. Theory, 35, n°4, 1989.

[K] V.Yu. KRATCHKOVSKII : " A decoding method for algebraic geometric codes", Proceeding of IXth Conference on coding theory and information transmission, Moscow-Odessa, 1988.(en russe).

[K-T]G.L. KATSMAN, M.A. TSFASMAN : "Spectra of algebraic-geometric codes", Prob. Peredachi Inf., 23, n°4, 1987; = Probl. Info. Trans., 23,1988, pp. 262-275.

[L] G. LACHAUD: "Les Codes Géométriques de Goppa", Séminaire Bourbaki, n°641, 1984-1985, pp.1-19.

[LB-R] D. LE BRIGAND, J.J. RISLER : " Algorithme de Brill-Noether et codes de Goppa", Bull. Soc. Math. France, 116,1988, pp 231-253.

[Pl] R. PELLIKAAN : "On a decoding algorithm for codes on maximal curves", to appear in IEEE Trans. Info. Theory.

[Pt] W.W. PETERSON, E.J. WELDON : Error correcting Codes , 2nd ed. , MIT Press , (1972).

[S-V] A.N. SKOROBOGATOV, S.G. VLADUTS : "On the decoding of Algebraic-Geometric Codes", preprint, Inst. for problems of Information Transmission, Moscow, to appear in IEEE Trans. Info. Theory.

[V] S.G. VLADUTS : "On the decoding of Algebraic-Geometric Codes over F_q for $q \geq 16$", preprint, Inst. for problems of Information Transmission, Moscow, to appear in IEEE Trans. Info. Theory.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399